

1 **CLAIMS**

2

3 1. A computer readable medium having stored thereon a data structure

4 that describes what types of binaries can be loaded into a process space for a

5 trusted application, the data structure comprising:

6 a first portion including data representing a unique identifier of the trusted

7 application;

8 a second portion including data indicating whether a particular one or more

9 binaries can be loaded into the process space for the trusted application; and

10 a third portion derived from the data in both the first portion and the second

11 portion by generating a digital signature over the first and second portions.

12

13 2. A computer readable medium as recited in claim 1, wherein the data

14 structure, when populated with data, is a manifest corresponding to the trusted

15 application, and wherein the unique identifier of the trusted application comprises:

16 a public key of a public-private key pair of a party that generates the

17 manifest;

18 an identifier of the party that generates the manifest; and

19 a version number of the manifest.

20

21

22

23

24

25

1 3. A computer readable medium as recited in claim 1, wherein the data
2 in the second portion comprises:

3 a list of one or more hashes of certificates that certify public keys which
4 correspond to private keys that were used to sign the certificates that correspond to
5 binaries that are authorized to execute in the process space.

6
7 4. A computer readable medium as recited in claim 3, wherein the data
8 in the second portion further comprises:

9 a list of one or more additional hashes of certificates that certify public keys
10 which correspond to private keys that were used to sign the certificates that
11 correspond to binaries that are not authorized to execute in the process space.

12
13 5. A computer readable medium as recited in claim 1, wherein the data
14 in the second portion comprises:

15 a list of one or more certificates that certify public keys which correspond
16 to private keys that were used to sign the certificates that correspond to binaries
17 that are authorized to execute in the process space.

18
19 6. A computer readable medium as recited in claim 5, wherein the data
20 in the second portion further comprises:

21 a list of one or more additional certificates that certify public keys which
22 correspond to private keys that were used to sign the certificates that correspond to
23 binaries that are not authorized to execute in the process space.

1 7. A computer readable medium as recited in claim 1, wherein the data
2 in the second portion comprises:

3 a list of one or more public keys which correspond to private keys that were
4 used to sign the certificates that correspond to binaries that are authorized to
5 execute in the process space.

6
7 8. A computer readable medium as recited in claim 7, wherein the data
8 in the second portion further comprises:

9 a list of one or more public keys which correspond to private keys that were
10 used to sign the certificates that correspond to binaries that are not authorized to
11 execute in the process space.

12
13 9. A computer readable medium as recited in claim 1, wherein the data
14 structure further comprises:

15 another portion that includes data representing a list of one or more export
16 statements that allow a secret associated with the trusted application to be exported
17 to another trusted application.

18
19 10. A computer readable medium as recited in claim 9, wherein the data
20 structure, when populated with data, is a manifest corresponding to the trusted
21 application, and wherein each of the one or more export statements comprises:

22 an identifier of the manifest;

23 an identifier of another manifest that corresponds to the trusted application
24 to which the secret is to be exported; and
25

1 a digital signature over both the identifier of the manifest and the identifier
2 of the other manifest.

3
4 **11.** A computer readable medium as recited in claim 10, wherein at least
5 one of the one or more export statements comprises:

6 an identification of a particular computing device on which the at least one
7 export statement is useable.

8
9 **12.** A computer readable medium as recited in claim 1, wherein the data
10 structure further comprises:

11 another portion that includes data representing a set of properties
12 corresponding to the data structure.

13
14 **13.** A computer readable medium as recited in claim 12, wherein the set
15 of properties includes:

16 whether the trusted application is debuggable.

17
18 **14.** A computer readable medium as recited in claim 12, wherein the set
19 of properties includes:

20 whether to allow an additional binary to be added to the process space after
21 the trusted application begins executing.

1 **15.** A computer readable medium as recited in claim 12, wherein the set
2 of properties includes:

3 whether to allow implicit upgrades to a higher version number.
4

5 **16.** A computer readable medium as recited in claim 1, wherein the data
6 structure further comprises:

7 another portion that includes data representing a list of entry points into the
8 executing trusted application.
9

10 **17.** A method of generating a new manifest to facilitate upgrading a
11 trusted application on a computing device to a new trusted application, the method
12 comprising:

13 receiving a request to upgrade the trusted application to the new trusted
14 application;

15 receiving one or more new components to be included in the new trusted
16 application; and

17 generating a manifest for the new trusted application, wherein the manifest
18 allows the one or more new components to be loaded on the computing device.
19

20 **18.** A method as recited in claim 17, further comprising:

21 digitally signing each of the one or more new components.
22
23
24
25

1 **19.** A method as recited in claim 17, further comprising:
2 making the manifest available to a trusted core of an operating system
3 executing on the computing device.

4
5 **20.** A method as recited in claim 17, wherein the manifest further
6 prevents one or more components of the trusted application from being loaded on
7 the computing device.

8
9 **21.** A method as recited in claim 17, wherein the manifest comprises:
10 a portion including data indicating whether a particular one or more
11 binaries can be loaded into a process space for the new trusted application.

12
13 **22.** A method as recited in claim 17, wherein the manifest comprises:
14 a first portion including data representing a unique identifier of the new
15 trusted application;
16 a second portion including data indicating whether a particular one or more
17 binaries can be loaded into a process space for the new trusted application;
18 a third portion derived from the data in both the first portion and the second
19 portion by generating a digital signature over the first and second portions;
20 a fourth portion that includes data representing a list of one or more export
21 statements that allow a secret associated with the new trusted application to be
22 exported to another trusted application; and
23 a fifth portion that includes data representing a set of properties
24 corresponding to the manifest.
25

1
2 **23.** A method as recited in claim 17, wherein the manifest includes an
3 identifier of the manifest, and wherein the identifier of the manifest includes:

4 a public key of a public-private key pair of a party that generates the
5 manifest;

6 an identifier of the party that generates the manifest; and

7 a version number of the manifest.
8

9 **24.** A method as recited in claim 23, wherein:

10 an original manifest corresponds to the trusted application;

11 the public key of the manifest is the same as a public key in an identifier
12 portion of the original manifest; and

13 the identifier of the party of the manifest is the same as an identifier, in the
14 original manifest, of the party that generated the original manifest.
15

16 **25.** A method as recited in claim 17, wherein generating the manifest
17 comprises:

18 adding, to the manifest, a list of one or more hashes of certificates that
19 certify public keys which correspond to private keys that were used to sign the
20 certificates that correspond to the one or more new components.
21
22
23
24
25

1 **26.** A method as recited in claim 17, wherein generating the manifest
2 comprises:

3 adding, to the manifest, a list of one or more certificates that certify public
4 keys which correspond to private keys that were used to sign the certificates that
5 correspond to the one or more new components.
6

7 **27.** A method as recited in claim 17, further comprising:
8 adding, to the manifest, an indication of each of one or more additional
9 components that cannot be executed in the process space.
10

11 **28.** A method as recited in claim 27, wherein adding, to the manifest, an
12 indication of each of one or more additional components that cannot be executed
13 in the process space, comprises:

14 adding, to the manifest, a list of one or more additional hashes of
15 certificates that certify public keys which correspond to private keys that were
16 used to sign the certificates that correspond to one or more additional components
17 that are not authorized to execute in the process space.
18

19 **29.** A method as recited in claim 27, wherein adding, to the manifest, an
20 indication of each of one or more additional components that cannot be executed
21 in the process space, comprises:

22 adding, to the manifest, a list of one or more additional certificates that
23 certify public keys which correspond to private keys that were used to sign the
24
25

1 certificates that correspond to one or more additional components that are not
2 authorized to execute in the process space.

3
4 **30.** A method comprising:
5 receiving a request to execute a process;
6 setting up a memory space for the process;
7 accessing a manifest corresponding to the process; and
8 limiting which of a plurality of binaries can be executed in the memory
9 space based on indicators, of the binaries, that are included in the manifest.

10
11 **31.** A method as recited in claim 30, wherein the manifest includes both
12 a list of a first set of indicators of binaries that can be executed in the memory
13 space and a list of a second set of indicators of binaries that cannot be executed in
14 the memory space.

15
16 **32.** A method as recited in claim 30, further comprising receiving, with
17 the request, the manifest.

18
19 **33.** A method as recited in claim 30, wherein limiting which of a
20 plurality of binaries can be executed in the memory space comprises:
21 loading a set of binaries into the memory space;
22 checking whether each binary in the set is consistent with the manifest; and
23 not allowing any of the binaries in the set to be executed unless all binaries
24 in the set are consistent with the manifest.
25

1
2 **34.** A method as recited in claim 33, wherein checking whether each
3 binary in the set is consistent with the manifest comprises checking, for each
4 binary, whether a certificate or a certificate hash corresponding to the binary is
5 included in a list of authorized binaries and is not included in a list of non-
6 authorized binaries.

7
8 **35.** A method as recited in claim 33, wherein limiting which of a
9 plurality of binaries can be executed in the memory space further comprises:

10 allowing the binaries in the set to be executed if all binaries in the set are
11 consistent with the manifest;

12 receiving a request to load an additional binary into the memory space;

13 checking whether the additional binary is consistent with the manifest; and

14 allowing the additional binary to be loaded into the memory space and
15 executed if it is consistent with the manifest, otherwise not loading the additional
16 binary into the memory space.

17
18 **36.** A method as recited in claim 30, wherein limiting which of a
19 plurality of binaries can be executed in the memory space comprises, for each of
20 the plurality of binaries:

21 checking whether the binary is consistent with the manifest;

22 loading the binary into the memory space if the binary is consistent with the
23 manifest; and
24
25

1 not loading the binary into the memory space if the binary is inconsistent
2 with the manifest.
3

4 **37.** A method as recited in claim 36, wherein checking whether the
5 binary is consistent with the manifest comprises checking whether a certificate or
6 a certificate hash corresponding to the binary is included in a list of authorized
7 binaries and is not included in a list of non-authorized binaries.
8

9 **38.** A method as recited in claim 36, wherein limiting which of a
10 plurality of binaries can be executed in the memory space further comprises:

11 receiving a request to load an additional binary into the memory space;
12 checking whether the additional binary is consistent with the manifest; and
13 allowing the additional binary to be loaded into the memory space and
14 executed if it is consistent with the manifest, otherwise not loading the additional
15 binary into the memory space.
16

17 **39.** A method as recited in claim 30, wherein memory space comprises a
18 virtual memory space.
19

20 **40.** A method as recited in claim 30, wherein the manifest comprises
21 data indicating whether a particular one or more binaries can be loaded into the
22 memory space for the process.
23
24
25

1 **41.** A method as recited in claim 30, wherein the manifest comprises:
2 a first portion including data representing a unique identifier of the process;
3 a second portion including data indicating whether a particular one or more
4 binaries can be loaded into the memory space for the process;
5 a third portion derived from the data in both the first portion and the second
6 portion by generating a digital signature over the first and second portions;
7 a fourth portion that includes data representing a list of one or more export
8 statements that allow a secret associated with the process to be exported to another
9 process; and
10 a fifth portion that includes data representing a set of properties
11 corresponding to the manifest.

12
13 **42.** A method as recited in claim 30, further comprising:
14 receiving, from the process, a request to securely store a secret, wherein the
15 request includes,
16 the secret,
17 a public key of a public-private key pair of a party that generated a
18 manifest for the process,
19 an identifier of the party, and
20 a set of one or more versions of the manifest that should be allowed
21 to retrieve the secret; and
22 having the secret encrypted.

1 **43.** A method as recited in claim 30, further comprising:
2 receiving, from the process, a request to securely store a secret, wherein the
3 request includes,
4 the secret, and
5 an identifier of one or more manifests that should be allowed to
6 retrieve the secret; and
7 having the secret encrypted.

8
9 **44.** A method as recited in claim 30, further comprising:
10 receiving, from the process, a request to retrieve a secret securely stored by
11 a previous process;
12 comparing the manifest identifier of the requesting process to a collection
13 of one or more manifest identifiers with which the secret was originally sealed;
14 and
15 determining whether to reveal the secret to the process based at least in part
16 on whether the manifest identifier of the requesting process and one of the
17 collection of manifest identifiers are the same.

18
19 **45.** A method as recited in claim 30, further comprising:
20 receiving encrypted data;
21 decrypting the data;
22 identifying a plurality of conditions in the data;
23 checking whether the manifest satisfies all of the plurality of conditions;
24 and
25

1 allowing the process to retrieve a secret in the encrypted data only if the
2 manifest satisfies all of the plurality of conditions.

3
4 **46.** A method as recited in claim 30, further comprising:
5 receiving encrypted data;
6 decrypting the data;
7 identifying one or more conditions in the data;
8 checking whether the data satisfies the one or more conditions; and
9 allowing the process to retrieve a secret in the encrypted data only if the
10 data satisfies the one or more conditions.

11
12 **47.** A method as recited in claim 30, further comprising:
13 receiving, from the process, a request to generate a digitally signed
14 statement; and
15 generating a digitally signed statement including an identifier of the
16 manifest corresponding to the process.

17
18
19 **48.** One or more computer readable media having stored thereon a
20 plurality of instructions that, when executed by one or more processors, causes the
21 one or more processors to:

22 set up a virtual memory space for a trusted application process;
23 obtain a manifest corresponding to the trusted application process;
24 identify, from the manifest, a plurality of binary indicators; and
25

1 restrict which of multiple binaries can be executed in the virtual memory
2 space based on the plurality of binary indicators.

3
4 **49.** One or more computer readable media as recited in claim 48,
5 wherein the instructions that cause the one or more processors to restrict which of
6 multiple binaries can be executed in the virtual memory space cause the one or
7 more processors to:

8 load a set of binaries into the virtual memory space;
9 check whether each binary in the set is consistent with the manifest; and
10 not allow any of the binaries in the set to be executed unless all binaries in
11 the set are consistent with the manifest.

12
13 **50.** One or more computer readable media as recited in claim 49,
14 wherein the instructions that cause the one or more processors to check whether
15 each binary in the set is consistent with the manifest cause the one or more
16 processors to check, for each binary, whether a certificate or a certificate hash
17 corresponding to the binary is included in a list of authorized binaries and is not
18 included in a list of non-authorized binaries.

1 **51.** One or more computer readable media as recited in claim 49,
2 wherein the instructions that cause the one or more processors to restrict which of
3 multiple binaries can be executed in the virtual memory space cause the one or
4 more processors to:

5 allow the binaries in the set to be executed if all binaries in the set are
6 consistent with the manifest;

7 receive a request to load an additional binary into the virtual memory space;

8 check whether the additional binary is consistent with the manifest; and

9 allow the additional binary to be loaded into the virtual memory space and
10 executed if it is consistent with the manifest, and otherwise not load the additional
11 binary into the virtual memory space.

12
13 **52.** One or more computer readable media as recited in claim 48,
14 wherein the instructions that cause the one or more processors to restrict which of
15 multiple binaries can be executed in the virtual memory space cause the one or
16 more processors to, for each of the multiple binaries:

17 check whether the binary is consistent with the manifest;

18 load the binary into the virtual memory space if the binary is consistent
19 with the manifest; and

20 not load the binary into the virtual memory space if the binary is
21 inconsistent with the manifest.

1 **53.** One or more computer readable media as recited in claim 52,
 2 wherein the instructions that cause the one or more processors to check whether
 3 the binary is consistent with the manifest cause the one or more processors to
 4 check whether a certificate or a certificate hash corresponding to the binary is
 5 included in a list of authorized binaries and is not included in a list of non-
 6 authorized binaries.

7
 8 **54.** One or more computer readable media as recited in claim 48,
 9 wherein the manifest comprises data indicating whether a particular one or more
 10 binaries can be loaded into the virtual memory space for the trusted application
 11 process.

12
 13 **55.** One or more computer readable media as recited in claim 48,
 14 wherein the manifest comprises:

15 a first portion including data representing a unique identifier of the trusted
 16 application process;

17 a second portion including data indicating whether a particular one or more
 18 binaries can be loaded into the virtual memory space for the trusted application
 19 process;

20 a third portion derived from the data in both the first portion and the second
 21 portion by generating a digital signature over the first and second portions;

22 a fourth portion that includes data representing a list of one or more export
 23 statements that allow a secret associated with the trusted application process to be
 24 exported to another trusted application process; and
 25

1 a fifth portion that includes data representing a set of properties
2 corresponding to the manifest.

3
4 **56.** One or more computer readable media having stored thereon a
5 plurality of instructions to implement a trusted core of a computing device that,
6 when executed by one or more processors of the computing device, causes the one
7 or more processors to:

8 receive, from a trusted agent executing on the computing device, a request
9 to securely store a secret, wherein the request includes,

10 the secret, and

11 an identifier of a manifest that should be allowed to retrieve the
12 secret; and

13 have the secret encrypted.

14
15 **57.** One or more computer readable media as recited in claim 56,
16 wherein the identifier comprises:

17 a public key of a public-private key pair of a party that generated the
18 manifest for the trusted agent;

19 an identifier of the party; and

20 a set of one or more versions of the manifest that should be allowed to
21 retrieve the secret.

1 **58.** One or more computer readable media having stored thereon a
2 plurality of instructions to implement a trusted core of a computing device that,
3 when executed by one or more processors of the computing device, causes the one
4 or more processors to:

5 receive, from a trusted application executing on the computing device, a
6 request to retrieve a secret securely stored by a previous trusted application
7 executing on the computing device;

8 compare a first manifest identifier of the trusted application to a second
9 manifest identifier corresponding to the previous trusted application; and

10 determine whether to reveal the secret to the trusted application based at
11 least in part on whether the first manifest identifier and the second manifest
12 identifier are the same.

13
14 **59.** One or more computer readable media as recited in claim 58,
15 wherein the trusted application is an upgraded version of the previous trusted
16 application.

17
18 **60.** One or more computer readable media as recited in claim 58,
19 wherein:

20 the first manifest includes a first public key of a first public-private key pair
21 of a party that generated the first manifest, an identifier of the party that generated
22 the first manifest, and a version indicator of the first manifest; and

1 the second manifest includes a second public key of a second public-private
2 key pair of a party that generated the second manifest, an identifier of the party
3 that generated the second manifest, and a version indicator of the second manifest.
4

5 **61.** One or more computer readable media as recited in claim 60,
6 wherein the instructions that cause the one or more processors to determine
7 whether to reveal the secret to the trusted application cause the one or more
8 processors to:

9 check whether the first public key is the same as the second public key and
10 whether the identity of the party that generated the first manifest is the same as the
11 identity of the party that generated the second manifest;

12 check whether the version indicator of the first manifest is the same as one
13 or more version indicators supplied by the previous trusted application when
14 securely storing the secret; and

15 refuse to reveal the secret to the trusted application if the checking indicates
16 any one or more of the following: the first public key is not the same as the
17 second public key, the identity of the party that generated the first manifest is not
18 the same as the identity of the party that generated the second manifest, the
19 version indicator of the first manifest is not the same as one or more version
20 indicators supplied by the previous trusted application when securely storing the
21 secret.
22
23
24
25

1 **62.** One or more computer readable media as recited in claim 58,
2 wherein the instructions that cause the one or more processors to determine
3 whether to reveal the secret to the trusted application further cause the one or more
4 processors to reveal the secret to the trusted application based at least in part on
5 whether an export certificate corresponding to the previous trusted application
6 identifies the first manifest as being authorized to retrieve the secret.

7
8 **63.** One or more computer readable media as recited in claim 62,
9 wherein the export certificate includes:

10 an identification of the first manifest;

11 an identification of the second manifest, wherein the second manifest was
12 digitally signed using a private key of a public-private key pair of a party that
13 generated the second manifest; and

14 a digital signature over the identification of the first manifest and the
15 identification of the second manifest, wherein the digital signature is generated
16 using the private key.

17
18 **64.** One or more computer readable media having stored thereon a
19 plurality of instructions to implement a trusted core of a computing device that,
20 when executed by one or more processors of the computing device, causes the one
21 or more processors to:

22 receive encrypted data;

23 decrypt the data;

24 identify a plurality of conditions in the data;

1 check whether a manifest associated with a trusted application process
2 satisfies all of the plurality of conditions; and

3 allow the trusted application process to retrieve a secret in the encrypted
4 data only if the manifest satisfies all of the plurality of conditions.

5
6 **65.** One or more computer readable media as recited in claim 64,
7 wherein:

8 the plurality of conditions include a public key of a public-private key pair
9 of a party that needs to have digitally signed the manifest; and

10 the instructions that cause the one or more processors to check whether the
11 manifest associated with the trusted application process satisfies all of the plurality
12 of conditions cause the one or more processors to check whether the manifest
13 includes the public key.

14
15 **66.** One or more computer readable media as recited in claim 64,
16 wherein:

17 the plurality of conditions include an identifier of a party that needs to have
18 generated the manifest; and

19 the instructions that cause the one or more processors to check whether the
20 manifest associated with the trusted application process satisfies all of the plurality
21 of conditions cause the one or more processors to check whether the manifest was
22 generated by the party.

1 **67.** One or more computer readable media as recited in claim 64,
2 wherein:

3 the plurality of conditions include one or more versions that the manifest
4 needs to be; and

5 the instructions that cause the one or more processors to check whether the
6 manifest associated with the trusted application process satisfies all of the plurality
7 of conditions cause the one or more processors to check whether the manifest is
8 one of the one or more versions.
9

10 **68.** One or more computer readable media as recited in claim 64,
11 wherein the plurality of instructions further cause the one or more processors to:

12 check whether the data satisfies one or more of the plurality of conditions;
13 and

14 allow the trusted application process to retrieve the secret only if the data
15 satisfies the one or more conditions.
16

17 **69.** One or more computer readable media having stored thereon a
18 plurality of instructions to implement a trusted core of a computing device that,
19 when executed by one or more processors of the computing device, causes the one
20 or more processors to:

21 receive, from a trusted application, a request to generate a digitally signed
22 statement; and

23 generate a digitally signed statement including an identifier of a manifest
24 corresponding to the trusted application.
25

1
2 70. One or more computer readable media as recited in claim 69,
3 wherein the plurality of conditions include a public key of a public-private key
4 pair of a party that digitally signed the manifest.
5

6 71. One or more computer readable media as recited in claim 69,
7 wherein the plurality of conditions include an identifier of a party that generated
8 the manifest.
9

10 72. One or more computer readable media as recited in claim 69,
11 wherein the plurality of conditions include one or more manifest version
12 indicators.
13

14 73. A computer readable medium having stored thereon a data structure
15 that allows a secret associated with a trusted application to be exported to another
16 trusted application, the data structure comprising:

17 a first portion including an identifier of a manifest associated with the
18 application;

19 a second portion including an identifier of a manifest associated with the
20 other application; and

21 a third portion derived from the identifiers in both the first portion and the
22 second portion by generating a digital signature over the first and second portions.
23
24
25

1 **74.** A computer readable medium as recited in claim 73, wherein the
2 data structure further comprises:

3 a fourth portion including an identification of a particular computing device
4 on which the data structure can be used to export the secret.

5
6 **75.** A computer readable medium as recited in claim 73, wherein the
7 second portion includes identifiers of a plurality of manifests associated with a
8 plurality of additional applications to which the secret can be exported.

9
10 **76.** A computer readable medium as recited in claim 75, wherein each
11 of the plurality of additional applications is a newer version of the other
12 application.